

[Sign in](#)
[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Local](#) [New!](#) [more »](#)

binary translation register renaming rollback re

[Advanced Search](#)  
[Preferences](#)

**Web** Results 1 - 10 of about 23,700 for **binary translation register renaming rollback recovery**. (0.26 seconds)

### Dynamic Binary Translation and Optimization

The target architecture for the DAISY **binary translation** system is a clustered

... The architected **register** state is obtained by **renaming** all speculative ...

doi.ieeecomputersociety.org/10.1109/12.931892 - [Similar pages](#)

### Changing Interaction of Compiler and Architecture

Processors can now implement **register renaming** and dynamic instruction ...

Dynamite: A Framework for Dynamic Retargetable **Binary Translation**, Tech. ...

doi.ieeecomputersociety.org/10.1109/2.642815 - [Similar pages](#)

[ [More results from doi.ieeecomputersociety.org](#) ]

### IP.com's Prior Art Database

A **binary translator** takes the source **binary** code as input, translates it and ...

A **rollback** and **recovery** method is used to map the source **registers** r1, r2, ...

www.priorartdatabase.com/IPCOM/000007121/ - [Similar pages](#)

### [PDF] Inherently Lower Complexity Architectures using Dynamic Optimization

File Format: PDF/Adobe Acrobat - [View as HTML](#)

gram semantics, **register renaming** is performed and. results are retired in-order

to the ... tion and Optimization Architecture", or "**Binary translation Op-** ...

www.research.ibm.com/people/ m/mikeg/papers/2002\_wced.pdf - [Similar pages](#)

### [PDF] IBM Research Report

File Format: PDF/Adobe Acrobat - [View as HTML](#)

gram semantics, **register renaming** is performed and. results are retired in-order

to the ... a target for **binary translation** and dynamic optimiza- ...

www.research.ibm.com/vliw/isca31/ RC22442%20-%20Inherently%20Lower%20Complexity%

20Architectures%20using%20... - [Similar pages](#)

### [PDF] An Instruction Set and Microarchitecture for Instruction Level ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

possibilities, both involve **binary translation**. One method. is to perform on-the-fly

hardware ... global **register rename** bandwidths per cycle. With a four- ...

www.ece.wisc.edu/~jes/papers/kimh\_ildp.pdf - [Similar pages](#)

### [PDF] Microsoft PowerPoint - ch15.ppt

File Format: PDF/Adobe Acrobat - [View as HTML](#)

Large visible **register** set – no hidden **rename** regs ... **rollback** instruction (?)

supports emulation ... **recover** latest committed emulated target **register** ...

www.cs.helsinki.fi/u/kerola/ tikra/s2003/luennot/ch15\_p6.pdf - [Similar pages](#)

### [PDF] IA-64 and Crusoe Architectures Ch 15

File Format: PDF/Adobe Acrobat - [View as HTML](#)

**rollback** instruction (?) supports emulation. • some or all of it can be in emulator

code. • **recover** latest committed emulated target **register** ...

www.cs.helsinki.fi/u/kerola/ tikra/s2002/luennot/ch15\_v.pdf - [Similar pages](#)

[PDF] [Dynamic Binary Translation and Optimization](#) Erik R. Altman Kemal ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

DAISY registers r36-r63 are used for renaming speculative. results during scheduling, and as ... Can a binary translation machine have generally better per- ...

[www.microarch.org/micro33/tutorial/m33-t-dcd.pdf](http://www.microarch.org/micro33/tutorial/m33-t-dcd.pdf) - [Similar pages](#)

[PDF] [Dynamic Optimization of Micro-Operations](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

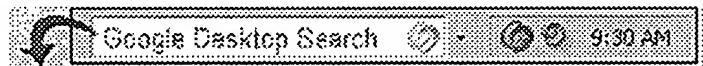
Dynamic and transparent binary translation. IEEE. Computer, 33:54 – 59, Mar. 2000. ... via a combined static/dynamic register renaming scheme. In ...

[www.crhc.uiuc.edu/~sjp/pub/hpca9.pdf](http://www.crhc.uiuc.edu/~sjp/pub/hpca9.pdf) - [Similar pages](#)

Try your search again on [Google Book Search](#)

Google

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)



Free! Instantly find your email, files, media and web history. [Download now.](#)

binary translation register renaming [Search](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2005 Google



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

Binary Translation register renaming rollback recovery



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

**Binary Translation register renaming rollback recovery**

Found 4,543 of 167,655

Sort results by

relevance

Display results

expanded form

[Save results to a Binder](#)[Search Tips](#)☐ Open results in a new windowTry an [Advanced Search](#)Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [RENO: A Rename-Based Instruction Optimizer](#)
 May 2005 **ACM SIGARCH Computer Architecture News , Proceedings of the 32nd Annual International Symposium on Computer Architecture ISCA '05,**  
 Volume 33 Issue 2
**Publisher:** IEEE Computer Society, ACM PressFull text available: [pdf \(256.23 KB\)](#) Additional Information: [full citation](#), [abstract](#)

RENO is a modified MIPS R10000 register renamer that uses map-table "short-circuiting" to implement dynamic versions of several well-known static optimizations: move elimination, common subexpression elimination, register allocation, and constant folding. Because it implements these optimizations dynamically, RENO can apply optimizations in certain situations where static compilers cannot. Several of RENO's component optimizations have been previously proposed as independent mechanisms. Unified ...

**2** [Novel ideas: Performance characterization of a hardware mechanism for dynamic optimization](#)

Brian Fahs, Satarupa Bose, Matthew Crum, Brian Slechta, Francesco Spadini, Tony Tung, Sanjay J. Patel, Steven S. Lumetta

 December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**
**Publisher:** IEEE Computer Society
 Full text available: [pdf \(1.31 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)  
[Publisher Site](#)

We evaluate the rePLay microarchitecture as a means for reducing application execution time by facilitating dynamic optimization. The framework contains a programmable optimization engine coupled with a hardware-based recovery mechanism. The optimization engine enables the dynamic optimizer to run concurrently with program execution. The recovery mechanism enables the optimizer to make speculative optimizations without requiring recovery code. We demonstrate that a rePLay configuration performing ...

**3** [Transient-fault recovery using simultaneous multithreading](#)

T. N. Vijaykumar, Irith Pomeranz, Karl Cheng

 May 2002 **ACM SIGARCH Computer Architecture News , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02 , Proceedings of the 29th annual international symposium on Computer**

**architecture ISCA '02**, Volume 30 Issue 2

**Publisher:** IEEE Computer Society, ACM Press

Full text available:  [pdf\(1.55 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)  
[Publisher Site](#)

We propose a scheme for transient-fault recovery called **Simultaneously and Redundantly Threaded processors with Recovery (SRTR)** that enhances a previously proposed scheme for transient-fault detection, called Simultaneously and Redundantly Threaded (SRT) processors. SRT replicates an application into two communicating threads, one executing ahead of the other. The trailing thread repeats the computation performed by the leading thread, and the values produced by the two threads are compared ...

#### 4 Checkpoint Processing and Recovery: Towards Scalable Large Instruction Window Processors

Haitham Akkary, Ravi Rajwar, Srikanth T. Srinivasan

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**

**Publisher:** IEEE Computer Society

Full text available:  [pdf\(419.08 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Large instruction window processors achieve high performance by exposing large amounts of instruction level parallelism. However, accessing large hardware structures typically required to buffer and process such instruction window sizes significantly degrade the cycle time. This paper proposes a novel Checkpoint Processing and Recovery (CPR) microarchitecture, and shows how to implement a large instruction window processor without requiring large structures thus permitting a high clock frequency. We fo ...




#### 5 Register integration: a simple and efficient implementation of squash reuse



Amir Roth, Gurindar S. Sohi

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

**Publisher:** ACM Press


Full text available:  [pdf\(154.98 KB\)](#)  
 [ps\(573.81 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)  
 [Publisher Site](#)

#### 6 Dynamic translation: The Transmeta Code Morphing™ Software: using speculation, recovery, and adaptive retranslation to address real-life challenges

James C. Dehnert, Brian K. Grant, John P. Banning, Richard Johnson, Thomas Kistler, Alexander Klaiber, Jim Mattson

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '03**

**Publisher:** IEEE Computer Society

Full text available:  [pdf\(988.25 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Transmeta's Crusoe microprocessor is a full, system-level implementation of the x86 architecture, comprising a native VLIW microprocessor with a software layer, the **Code Morphing Software (CMS)**, that combines an interpreter, dynamic binary translator, optimizer, and runtime system. In its general structure, CMS resembles other binary translation systems described in the literature, but it is unique in several respects. The wide range of PC workloads that CMS must handle gracefully in real ...

**Keywords:** binary translation, dynamic optimization, dynamic translation, emulation,

self-modifying code, speculation

## 7 An analysis of a resource efficient checkpoint architecture



Haitham Akkary, Ravi Rajwar, Srikanth T. Srinivasan

December 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**,

Volume 1 Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(757.69 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Large instruction window processors achieve high performance by exposing large amounts of instruction level parallelism. However, accessing large hardware structures typically required to buffer and process such instruction window sizes significantly degrade the cycle time. This paper proposes a novel checkpoint processing and recovery (CPR) microarchitecture, and shows how to implement a large instruction window processor without requiring large structures thus permitting a high clock frequency ...

**Keywords:** Computer architecture, checkpoint architecture, high-performance computing, scalable architecture

## 8 Transient-fault recovery for chip multiprocessors



Mohamed Gomaa, Chad Scarbrough, T. N. Vijaykumar, Irith Pomeranz

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture ISCA '03**, Volume 31 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(370.75 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

To address the increasing susceptibility of commodity chip multiprocessors (CMPs) to transient faults, we propose Chiplevel Redundantly Threaded multiprocessor with Recovery (CRTR). CRTR extends the previously-proposed CRT for transient-fault detection in CMPs, and the previously-proposed SRTR for transient-fault recovery in SMT. All these schemes achieve fault tolerance by executing and comparing two copies, called leading and trailing threads, of a given application. Previous recovery schemes ...

## 9 Increasing Register File Immunity to Transient Errors



Gokhan Memik, Mahmut T. Kandemir, Ozcan Ozturk

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1**

**Publisher:** IEEE Computer Society

Full text available: [pdf\(162.75 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Transient errors are one of the major reasons for system downtime in many systems. While prior research has mainly focused on the impact of transient errors on datapath, caches and main memories, the register file has largely been neglected. Since the register file is accessed very frequently, the probability of transient errors is high. In addition, errors in it can quickly spread to different parts of the system, and cause application crash or silent data corruption. This paper addresses the r ...

## 10 Using speculative retirement and larger instruction windows to narrow the performance gap between memory consistency models



Parthasarathy Ranganathan, Vijay S. Pai, Sarita V. Adve

June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

**Publisher:** ACM Press



Full text available: [pdf\(1.83 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

# 11 Register file and memory system design: Reducing register ports for higher speed and lower energy

Il Park, Michael D. Powell, T. N. Vijaykumar

November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

**Publisher:** IEEE Computer Society Press

Full text available:  [pdf\(1.28 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)  
[Publisher Site](#)

The key issues for register file design in high-performance processors are access time and energy. While previous work has focused on reducing the number of registers, we propose to reduce the number of register ports through two proposals, one for reads and the other for writes. For reads, we propose bypass hint to reduce register port requirements by avoiding unnecessary register file reads for cases where values are bypassed. Current processors are unable to avoid these unnecessary reads due ...

# 12 Fingerprinting: bounding soft-error detection latency and bandwidth



Jared C. Smolens, Brian T. Gold, Jangwoo Kim, Babak Falsafi, James C. Hoe, Andreas G. Nowatzky

October 2004 **ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating Systems Review , Proceedings of the 11th international conference on Architectural support for programming languages and operating systems ASPLOS-XI**, Volume 39 , 32 , 38 Issue 11 , 5 , 5

**Publisher:** ACM Press

Full text available:  [pdf\(229.65 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Recent studies have suggested that the soft-error rate in microprocessor logic will become a reliability concern by 2010. This paper proposes an efficient error detection technique, called *fingerprinting*, that detects differences in execution across a dual modular redundant (DMR) processor pair. Fingerprinting summarizes a processor's execution history in a hash-based signature; differences between two mirrored processors are exposed by comparing their fingerprints. Fingerprinting tightly ...

**Keywords:** backwards error recovery (BER), dual modular redundancy (DMR), error detection, soft errors


# 13 An out-of-order execution technique for runtime binary translators



Bich C. Le

October 1998 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 32 , 33 Issue 5 , 11

**Publisher:** ACM Press



Full text available:  [pdf\(1.04 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A dynamic translator emulates an instruction set architecture by translating source instructions to native code during execution. On statically-scheduled hardware, higher performance can potentially be achieved by reordering the translated instructions; however, this is a challenging transformation if the source architecture supports precise exception semantics, and the user-level program is allowed to register exception handlers. This paper presents a software technique which allows a translator ...

Superscalar design: Cherry: checkpointed early resource recycling in out-of-order microprocessors

José F. Martínez, Jose Renau, Michael C. Huang, Milos Prvulovic, Josep Torrellas  
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

**Publisher:** IEEE Computer Society Press

Full text available:  pdf (1.40 MB)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)  
[Publisher Site](#)

This paper presents *Checkpointed Early Resource RecYcling (Cherry)*, a hybrid mode of execution based on ROB and checkpointing that decouples resource recycling and instruction retirement. Resources are recycled early, resulting in a more efficient utilization. Cherry relies on state checkpointing and rollback to service exceptions for instructions whose resources have been recycled. Cherry leverages the ROB to (1) not require in-order execution as a fallback mechanism, (2) allow memory re ...

15 Toward kilo-instruction processors

 Adrián Cristal, Oliverio J. Santana, Mateo Valero, José F. Martínez  
December 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 1 Issue 4


**Publisher:** ACM Press

Full text available:  pdf (1.16 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The continuously increasing gap between processor and memory speeds is a serious limitation to the performance achievable by future microprocessors. Currently, processors tolerate long-latency memory operations largely by maintaining a high number of in-flight instructions. In the future, this may require supporting many hundreds, or even thousands, of in-flight instructions. Unfortunately, the traditional approach of scaling up critical processor structures to provide such support is impractical ...

**Keywords:** Memory wall, instruction-level parallelism, kilo-instruction processors, multichunkpointing

16 Enhancing software reliability with speculative threads

 Jeffrey Oplinger, Monica S. Lam  
October 2002 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 37 , 36 , 30 Issue 10 , 5 , 5

**Publisher:** ACM Press



Full text available:  pdf (1.47 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper advocates the use of a monitor-and-recover programming paradigm to enhance the reliability of software, and proposes an architectural design that allows software and hardware to cooperate in making this paradigm more efficient and easier to program. We propose that programmers write monitoring functions assuming simple sequential execution semantics. Our architecture speeds up the computation by executing the monitoring functions speculatively in parallel with the main computation. For ...

17 Novel ideas: Skipper: a microarchitecture for exploiting control-flow independence

Chen-Yong Cher, T. N. Vijaykumar  
December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

**Publisher:** IEEE Computer Society

Full text available:  pdf (1.51 MB)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Publisher Site

Although modern superscalar processors achieve high branch prediction accuracy, certain branches either are inherently difficult to predict or incur destructive interference in prediction tables, causing significant performance loss due to mispredictions. We propose a novel microarchitecture, called Skipper, to handle such difficult branches by exploiting control-flow independence. Previous approaches to handling difficult branches, one way or another, amount to executing incorrect instructions, ...

18 The taser intrusion recovery system

Ashvin Goel, Kenneth Po, Kamran Farhadi, Zheng Li, Eyal de Lara

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5**Publisher:** ACM PressFull text available: pdf(346.32 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Recovery from intrusions is typically a very time-consuming operation in current systems. At a time when the cost of human resources dominates the cost of computing resources, we argue that next generation systems should be built with automated intrusion recovery as a primary goal. In this paper, we describe the design of Taser, a system that helps in selectively recovering legitimate file-system data after an attack or local damage occurs. Taser reverts tainted, i.e. attack-dependent, file-syst ...

**Keywords:** file systems, intrusion analysis, intrusion recovery, snapshots19 Selective eager execution on the PolyPath architecture

Artur Klauser, Abhijit Paithankar, Dirk Grunwald

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3**Publisher:** IEEE Computer Society, ACM PressFull text available: pdf(1.57 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)  
Publisher Site

Control-flow misprediction penalties are a major impediment to high performance in wide-issue superscalar processors. In this paper we present *Selective Eager Execution* (SEE), an execution model to overcome mis-speculation penalties by executing both paths after diffident branches. We present the micro-architecture of the *PolyPath* processor, which is an extension of an aggressive superscalar, out-of-order architecture. The PolyPath architecture uses a novel instruction tagging and ...

20 Using Dynamic Binary Translation to Fuse Dependent Instructions

Shiliang Hu, James E. Smith

March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '04****Publisher:** IEEE Computer SocietyFull text available: pdf(240.50 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Instruction scheduling hardware can be simplified and easily pipelined if pairs of dependent instructions are fused so they share a single instruction scheduling slot. We study an implementation of the x86 ISA that dynamically translates x86 code to an underlying ISA that supports instruction fusing. A microarchitecture that is co-designed with the fused instruction set completes the implementation. In this paper, we focus on the dynamic binary translator for such a co-designed x86 virtual machine. The dy ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)